

飞浪脚本零起点入门系列(五)

MAXScript 的数组, 循环语句和重命名脚本写法

作者: 飞浪

声明:本教程为 **CG++** 原创,转载请注明出处,谢谢合作:)

本节关键词:

Array Values 数组 for loop selection objects uniqueness

Array Values 数组

数组,即有序数据的集合。这个数据可以是任何数据类型,甚至表达式。

语法:#(, ...)

#() 一空数组

例子: #(1, 2, "cgplusplus", 5+6, 123 as float) 注意, 括号, 引号等字符要在英文半角状态下输入。

那么,我们用数组就可以把一些相干或者不相干的数据都统一起来,放到一起,这样操作起来非常方便。比如说刚才那个数组,你可以一个一个的赋值:

a=1, b=2, c="cgplusplus"... , 然后调用。如果用数组,可以很方便:

数组引用[],其中就是数组,就是序号。数组调用的序号用中括号[]括起来。比如如果想调用第5个的值是这样:

#(1, 2, "cgplusplus", 5+6, 123 as float)[5]。这样还是很麻烦啊,先赋值再调用,就简单多了。

a = #(1, 2, "cgplusplus", 5+6, 123 as float)

a[1]就是第一个值,a[2]就是第二个值,依次类推。但是序号不能超出数组数据的最大数目,如a[6]就是错误引用,返回错误代码。其他相关操作讲解:

.count : Integer 一数组元素的个数,只能读取。Integer表明执行此命令返回整数。如a.count返回5,说明这个数组一共有5个元素。

[] 一修改某元素的值。如:a[1] = 100,再看一下数组a变成了#(100, 2, "cgplusplus", 11, 123.0)即第一个值发生了改变。

as array 一把集合转变成数组。如:b = lights as array 即把场景所有灯光存储为数组并赋值给b。

append 一向数组里面增加元素,如:append a "cg++" 即向数组a里面增加元素"cg++",得到新数组,增加的新元素排在数组最后面,这样数组a变成了:#(100, 2, "cgplusplus", 11, 123.0, "cg++"),a.count也返回了6。

deleteItem 一删除数组某个元素。如:deleteItem a 4 即把数组a的第4个元素删除,数组序号自动往前移动一位。这时我们得到:

#(100, 2, "cgplusplus", 123.0, "cg++")

findItem 一寻找数组里面是否有某元素。如:findItem a "cgplusplus",如果找到了某元素,返回第一次找到此元素的序号(因为数组里面的元素是可重复的),如果找不到,则返回0。

数组还有其他一些操作语句，这里一次不讲太多，在以后的实例中本人会详细讲解，这里只是想让大家对 maxscript 数组有个大致了解。

循环语句

for loop [for loop 在线手册](#)

for 循环可以逐个计算一系列数值，集合，数组等等，刚刚说的数组正好派上了用场。

语法：for (in | =) (do | collect)

是循环中的每一个单独的变量，是循环的源数值，是表达式，|号左右表示可选项。看实例吧：

例一：for i in 1 to 10 do print i --输出 1 到 10

print 输出函数，后面接所有类型数值，如：print selection[1].name --输出选择的第一个物体的名字

刚才那句执行的顺序是：首先赋值 i=1，然后执行 print i 就输出 1，再 i 进行自加，变成了 2，又执行 print i 输出 2... 一直到 i=10 输出 10 才算完。这样就进行了循环。而且是每隔 1 个数值输出一个值，下面来扩展：

例二：for i in 1 to 10 by 2 do print i --每隔两个数来输出 i，于是我们就得到了奇数 1, 3, 5, 7, 9。by 就是控制每隔多少数值再执行 do 后面的表达式。默认不用 by，则都是每隔 1 个数值执行一下。下面来看这句程序：

例三：for i in selection do i.name += "new"

selection 即所有选择的物体，相当于美元符号 "\$"，但是用法有时候不同。

selection.count 即选择物体数量，selection 即选择中的第 i 个物体，selection 类似于一个数组。上面那句的意思是，把所有选择物体的名字后面加一个 "new"，你再看看你选择的物体名称。上面那句还可写成：

for i in \$ do i.name += "new" 在程序中建议尽量使用 selection，因为如果你没有选择物体，那么这句就出错。

这两句是直接访问数组元素，从第一个开始，而不像前面的 for i in 1 to 10 ... 还要数元素序号。这一句就跟平常简单的英文会话一样，可以翻译成：如果 i(变量)在 selection (选择物体) 里面，那么就 do (做) 什么什么。

例四：for i in lights do i.on = false 这句意思是说，把场景所有灯光关掉，i.on 是开，i.off 是关，跟日常开关一样。有人也许会问，你怎么知道所有灯光是用 lights 表示呢？MAX 工具栏上有个过滤器，想必大家都用过，里面有 All, Geometry, Shapes, Lights, Cameras, Helpers 等，比如你想改所有 for i in cameras do ... 就是对所有摄像机操作了。但是 All 在脚本里并不是所有物体。再回来看例四，你打个有目标点的灯光试试，再用那句就会出错了，因为 lights 把灯光目标点也算在内，而目标点是没有开关的啊，那怎么办呢？这就要把它过滤掉。再来看，

for i in lights where classof i != targetobject do i.on = false 这里面加了一句 where classof i != targetobject 这句相当于一个过滤器，当变量 i 在 lights 里面逐个执行的时候，先要过这一关，如果 classof i != targetobject 返回 true 就继续执行 do 后面的语句，如果返回 false 就直接跳到下一个元素。这里用到了 maxscript 的比较运算符 !=，maxscript 比较运算符即把两个数据进行比较，返回 true 或者 false，所有比较运算符有：

比较运算在线手册：

http://www.cgplusplus.com/online-reference/maxscript-reference/source/comparison_expressions.htm

假设 a 和 b 是分别要比较的两个数据，下面即把 a, b 进行比较：

a == b --相等，如果相等就返回 true，不相等就返回 false，下面道理相同

a != b --不等于

a > b --a 大于 b

a < b --a 小于 b

a >= b --a 大于或等于 b

a <= b --a 小于或等于 b

例如：1 > 2 返回 false, 1 >= 1 返回 true

这里还用到了类别运算 classof, classof 是用来判断一个 max 节点的类别的，用法是直接 classof + 节点，如果我们不知道一个物体的类别，就直接选择它然后 classof \$, 即判断选择物体的类别，比如上面我本来不知道灯光目标点的类别，于是我选择一个目标点，再运行 classof \$, 就返回了它的类别 targetobject, 这样，就可以过滤掉所有类别是 targetobject 的物体，以免出错。

下面来讲一个重命名脚本，我们看到一些脚本里都有重命名功能，其实这个功能要写出来非常的简单，简单到让人惊讶。上面已经牛刀小试，例三：for i in selection do i.name += "new" 就是一个重命名功能，是在所有选择物体后面加个 "new"。真正的重命名脚本如下：

例五：for i in objects do i.name = uniqueName "cgplusplus"

这里无非是用到一 uniqueName, [uniqueName 在线手册](#)：最下面有它的功能。

它的原理跟你创建物体的命名一样，把物体以所给字符串为基础进行命名，后面跟数字序号。比如我执行上面那两句之后，再看场景中的物体名字分别变成了

cgplusplus01, cgplusplus02, cgplusplus03...

objects --即场景所有物体，objects.count 即场景所有物体数量。objects 即场景中的第 i 个物体。objects 也类似一个数组。

再把上面那句改一下：**例六：**

newName = "cgplusplus"

for i in objects do i.name = newName + "_" + i.name

这里就是将所有物体 objects 的名称加个前缀 "cgplusplus_"，这个下划线 "_" 并没有先赋值，这里是想说，newName + "_" + i.name 这一句可以是任意的字符串，可以用加号进行调整。上面是先把名称 "cgplusplus" 赋给变量 newName 再调用它，建议大家以后也这样做先赋值再调用，因为这样程序修改起来非常方便。下面进行扩展：

例七：for i in cameras do i.name = uniqueName "MyCamera" --把场景摄像机名字改成 MyCamera01, MyCamera02, MyCamera03...

例八：for b in \$box* do b.name = uniqueName "MyBox" --把场景所有名称以 "box" 开头的物体重命名，\$box 就是名称为 box 的物体，后面加一个通配符 "*" 就表示任意字符，\$box* 就是指名称以 "box" 开头的物体。注意变量 b 前后要一致，小心都写成了 i。

例九：for vl in lights where classof vl == VRayLight do vl.subdivs = 12 --把场景中所有 VRayLight 的细分调到 12

例十: `for g in geometry do g.wirecolor = red` --把所有几何体的线框色改成红色, 颜色常量请查看第二节。

我想这例子够多了, 只要你肯动脑筋, 就可以在工作和学习中使用一句简单的语句, 就实现很强大的功能。循环语句还有 `while...do` 没讲, 以后用到再讲吧。

这些讲的都是用 `maxscript listener` 执行脚本, 可能有些人想学习怎么样把他们写成界面, 按钮然后来使用吧? 别急, 后面我会讲到, 把这几节基础内容搞清楚了, 后面学起来简单很多。写教程的确很费脑筋, 希望我的工作没有白费, 对想学习 `max` 脚本而无处入门的同仁给以光亮。也希望大家继续支持我, 谢谢。

本节原帖地址: <http://www.cgplusplus.com/bbs/viewthread.php?tid=88>

本节结束。